

INTEGRACION DE APLICACIONES EN ENTORNO WEB

Ing. Esp. Romero, María Soledad
Ing. Romero, Rubén Aníbal

SERVICIOS REST (CONT.)

Integración de aplicaciones entornos web

API Rest - Niveles

- Nivel 1 - Uso correcto de URIs
- Nivel 2 - Uso correcto de HTTP.
- Nivel 3 - Implementar Hypermedia.

Nivel 0 - Protocolo de transporte

- Se utiliza HTTP como un protocolo de transporte.

Nivel 1 - Uso correcto de URIs

Reglas básicas:

- ❑ Los nombres de URI no deben implicar una acción, por lo tanto debe evitarse usar verbos en ellos.
- ❑ Deben ser únicas, no debemos tener más de una URI para identificar un mismo recurso.
- ❑ Deben ser independiente de formato.
- ❑ Deben mantener una jerarquía lógica.
- ❑ Los filtrados de información de un recurso no se hacen en la URI.

Nivel 1 - Ejemplos

GET /facturas

Nos permite acceder al listado de facturas

POST /facturas

Nos permite crear una factura nueva

GET /facturas/123

Nos permite acceder al detalle de una factura

PUT /facturas/123

Nos permite editar la factura, sustituyendo la totalidad de la información anterior por la nueva.

Nivel 1 - Ejemplos



DELETE /facturas/123

Nos permite eliminar la factura

PATCH /facturas/123

Nos permite modificar cierta información de la factura, como el número o la fecha de la misma.

Nivel 2 - Uso correcto de HTTP

- Métodos HTTP
- Códigos de estado
- Aceptación de tipos de contenido

Métodos HTTP

- GET: Para consultar y leer recursos
- POST: Para crear recursos
- PUT: Para editar recursos
- DELETE: Para eliminar recursos.
- PATCH: Para editar partes concretas de un recurso.

Códigos de estado



Aceptación de tipos de contenido

- **Header Accept:** HTTP nos permite especificar en qué formato queremos recibir el recurso, pudiendo indicar varios en orden de preferencia, para ello utilizamos el header Accept.
- **Content-Type:** En la respuesta, se devolverá el header Content-Type, para que el cliente sepa qué formato se devuelve.

Nivel 3 - Implementar Hypermedia



Asociaciones

- Obtener información contextual del objeto que está detrás. Ej: `/api/Clientes/123/Facturas`
- Para cosas más específicas usar los parámetros de entrada. Ej: `/api/Clientes?estado=activo`

Escenarios Consumo de APIs

- Consumo con un Cliente HTTP:
 - Tipos de Aplicaciones
 - Aplicaciones Escritorio.
 - Mobile.
 - Web. Invocaciones desde un WebServer (IIS, Apache, Tomcat)

Escenarios Consumo de APIs

- Ajax. Javascript.
 - Nativo cualquier Browser (Chrome, IE, Firefox, etc.).
 - Frameworks
 - JQuery.
 - AngularJs
 - KnockOut.

HTTP Status Codes

- POST: 201 (created), 400 (bad request), 500 (internal server error)
- GET: 200 (ok), 404 (not found), 400, 500
- DELETE: 204 (no content), 404, 400, 500
- PUT: 200, 404, 400, 500
- PATCH: 200, 404, 400, 500
- General: 401 (unauthorized), 403 (forbidden), 405 (method not allowed)

Formato Respuesta – Header Accept

http://localhost:8796/api/t + No environment

GET http://localhost:8796/api/test2/5 Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Generate Code

<input checked="" type="checkbox"/>	Accept	application/json	⋮ ×	Presets
	key	value	Bulk Edit	

Attribute Routing

```
[Route("api/courses/{id:int}")]  
public string Get(int id)  
{  
    return "value";  
}
```

IHttpActionResult

```
[Route("{id:int}")]
public IHttpActionResult GetCourse(int id)
{
    Learning.Data.LearningContext ctx = null;
    Learning.Data.ILearningRepository repo = null;
    try
    {
        ctx = new Learning.Data.LearningContext();
        repo = new Learning.Data.LearningRepository(ctx);

        var course = repo.GetCourse(id, false);
        if (course != null)
        {
            return Ok<Learning.Data.Entities.Course>(course);
        }
        else
        {
            return NotFound();
        }
    }
    catch (Exception ex)
    {
        return InternalServerError(ex);
    }
    finally
    {
        ctx.Dispose();
    }
}
```

Configuración IIS. IISExpress

- Path configuración IISExpress:
 - C:\Users\[usuario]\Documents\IISExpress\config\applicationhost.config
- Habilitar el consumo de los verbos PUT, DELETE:
 - **Path XML:**
/configuration/location/system.webServer/handlers

CORS. Cross-origin resource sharing.

- Intercambio de recursos de origen cruzado.
- Mecanismo que permite a los recursos restringidos (por ejemplo, tipos de letra) de una página web ser solicitados desde otro dominio fuera del dominio desde el que se originó el recurso.

Links

- **Resolve 404 in IIS Express for PUT and DELETE Verbs:**
<http://stevemichelotti.com/resolve-404-in-iis-express-for-put-and-delete-verbs/>
- **CORS:**
<http://enable-cors.org/>
- **API Rest Internet**
<http://jsonplaceholder.typicode.com/>
<http://jsonplaceholder.typicode.com/users/>

Links

- **Api Rest**

<http://asiermarques.com/2013/conceptos-sobre-apis-rest/>

<https://martinfowler.com/articles/richardsonMaturityModel.html>

<https://elbauldelprogramador.com/buenas-practicas-para-el-diseno-de-una-api-restful-pragmatica/>